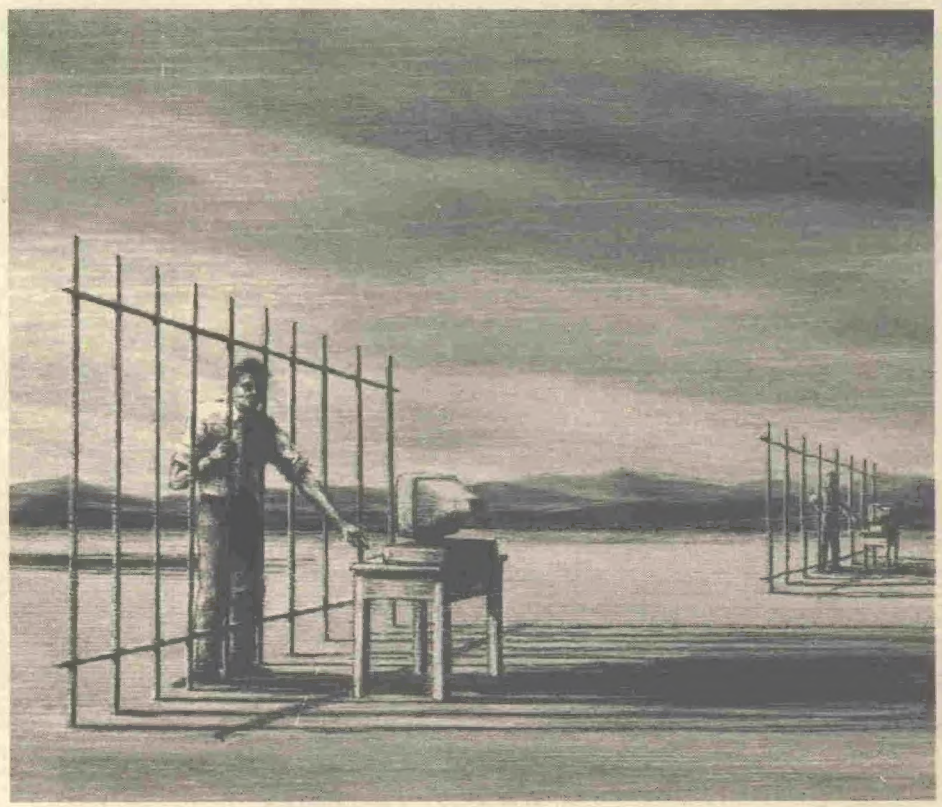
การรักษาความปลอดภัยระบบยูนิกซ์

สร้างความปลอดภัยให้กับระบบยูนิกซ์ให้หลุดพ้นจากการรบกวนเมื่อปฏิบัติงาน



แรกเริ่มเดิมทีระบบปฏิบัติการยูนิกซ์ถูกออกแบบและสร้างขึ้นโดยโปรแกรมเมอร์เพื่อให้โปรแกรมเมอร์ด้วยกันใช้ ปกติแล้วทีมโปรแกรมเมอร์จะร่วมกันทำงานโปรเจ็กต์หนึ่ง ๆ จะมีการใช้ไฟล์ร่วมกัน ซึ่งก็หมายถึงต้องการระบบที่มีการติดต่อร่วมกันทำงานได้ง่าย โดยไม่เน้นความเป็นส่วนตัวและไม่อยากให้ระบบรักษาความปลอดภัยมาทำเสียเวลา

พอยูนิกซ์เริ่มออกมาใช้ภายนอกห้องวิจัยที่เป็นจุดกำเนิดของมัน ก็มาใช้ในห้องวิจัยอื่น ๆ ซึ่งก็ยังไม่คำนึงถึงเรื่องความปลอดภัยมากนัก จนกระทั่งช่วงต้นทศวรรษ 1980 ที่หลาย ๆ มหาวิทยาลัยเริ่มเอาออกมาใช้ในศูนย์คอมพิวเตอร์ หน่วยงานทั้งธุรกิจและองค์กรรัฐบาลเริ่มสัมผัส ตอนนี้เป้าหมายทางด้านความปลอดภัยก็มีความจำเป็นขึ้นมาก อย่างน้อยก็ต้องไม่ให้นักเรียนลอกการบ้าน (ทางคอมพิวเตอร์) กัน หน่วยงานธุรกิจยิ่งต้องการความปลอดภัยอย่างมากสำหรับข้อมูลของงานทางบัญชีและคงคลัง เป็นต้น

เมื่อระบบปฏิบัติการยูนิกซ์เริ่มเป็นที่นิยมกันมากขึ้นและขนาดโตขึ้น ระบบรักษาความปลอดภัยก็กลายเป็นเรื่องซับซ้อนขึ้น ยิ่งเมื่อต่อเครื่องที่ใช้ยูนิกซ์หลาย ๆ เครื่องเข้าด้วยกันเป็นระบบเครือข่าย การเข้าสู่อินเตอร์เน็ตซึ่งเป็นการเชื่อมโยงระบบเครือข่ายหลาย ๆ กลุ่มเข้าด้วยกันหมด ระบบการ remote login การใช้ไฟล์ข้ามเครื่องกันได้ ฯลฯ เหล่านี้ก็เป็นการเพิ่มช่องว่างหรือจุดอ่อนด้านความปลอดภัยแก่เครื่องที่ใช้ยูนิกซ์ได้โดยไม่ต้องสงสัย

INTERNET WORM

ประมาณปลายปี 1988 โปรแกรมที่สามารถแพร่พันธุ์ตัวเองได้ที่เรียกกันว่า "WORM" ได้ถูกสร้างขึ้นและปล่อยลงในอินเตอร์เน็ต เพียงชั่วข้ามคืน โปรแกรมนี้ได้แพร่กระจายตัวเองจากเครื่องหนึ่งไปสู่เครื่องอื่น ๆ และทำให้เครื่องที่ถูกติดเชื้อนี้ทำงานบางอย่างซึ่งเป็นการแย่งไม่ให้เครื่องไปทำงานอื่นตามปกติที่ควรจะทำ หรือที่เรียกว่าโหลด (load) เครื่องนั่นเอง ทำให้เครื่องที่ติดนั่นเสียหาย

ที่จริงแล้ว WORM ไม่ได้ทำงานไฟล์ ขโมยไฟล์หรือข้อมูลสำคัญหรอก เพียงแต่มันไปรบกวนการทำงานของเครื่องที่มันเข้าไปอยู่เท่านั้น ต่างกับโปรแกรมพวกที่เรียกว่าไวรัสซึ่งมักจะถูกเขียนขึ้นมาเพื่อไปทำลายบางสิ่งบางอย่าง แต่ถึงอย่างไร WORM พวกนี้ก็ได้สร้างความวุ่นวายมากทีเดียว จนหน่วยงานสำคัญ ๆ หลายแห่งได้ถอดเครื่องตัวเองออกจากอินเตอร์เน็ตชั่วคราวเลยทีเดียว เช่น Goddard Space Flight Center ของ NASA และศูนย์วิจัย US. Army Ballastic เป็นต้น และถึงกับมีการสั่งให้ตัดการติดต่อระหว่างเครือข่าย MILNET และ ARPANET ออกจากกันเกือบ 24 ชั่วโมงเพื่อความปลอดภัยของคอมพิวเตอร์ทางการทหารด้วย

เหตุการณ์ครั้งนี้ตกเป็นข่าวใหญ่ที่ลงในหนังสือพิมพ์และนิตยสารมากมายรวมทั้ง New York Times, Wall Street Journal, ข่าว CNN และวิทยุ NPR (National Public Radio) และยังทำให้ระบบปฏิบัติการยูนิกซ์กับเน็ตเวอร์กตกเป็นเป้าของปัญหาความปลอดภัยของระบบคอมพิวเตอร์มาโดยตลอด แล้วผลลงเอยของผู้เขียน WORM ตัวนี้ก็คือถูกปรับและติดคุกในที่สุด

แล้ว WORM นี้ทำงานอย่างไร มันหาช่องว่างที่มีอยู่น้อยนิดในบางเครื่องที่ใช้ยูนิกซ์อย่างเช่น คำสั่ง finger ที่ให้ผู้ใช้ดูข้อมูลเกี่ยวกับผู้ใช้อื่น ๆ ในเครื่องหรือในเครือข่ายได้ เช่น ดูชื่อจริง ที่อยู่ เป็นต้น โปรแกรม finger นี้ เวอร์ชันแรก ๆ มีข้อบกพร่องหรือบักอยู่ที่ทำให้เกิดอาการบัฟเฟอร์เกินได้ เจ้า WORM นี้ก็ได้ใช้ช่องว่างนี้ฝากข้อมูลบางอย่างเข้าไปไว้ในสแต็ก (stack) และนั่นก็สามารถนำไปสู่การแพร่ตัวเองเข้าไปในเครื่องอื่น ๆ ได้

นอกจากโปรแกรม finger แล้ว โปรแกรม sendmail ซึ่งใช้ส่งจดหมายภายในเครือข่าย ก็มีจุดอ่อนคือออปชัน debug ซึ่งถ้าเปิดใช้แล้วก็สามารถที่จะส่งคำสั่งไปทำงานได้นอกเหนือจากข้อมูล ปกติแล้วออปชันเหล่านี้ควรจะปิดไว้ แต่สมัยก่อนก็มีการเผลอเปิดไว้ในบางเครื่องในเครือข่าย

สายลับ

ก่อนหน้าที่จะมีโปรแกรมพวก WORM นี้ ช่วงปี 1986 ที่ศูนย์วิจัยแห่งหนึ่งในมหาวิทยาลัย UC. Berkeley ถูกบุกรุกทางคอมพิวเตอร์ Clifford Stoll ผู้เป็น system administrator หรือผู้ดูแลระบบเครื่องนั้นก็ได้เฝ้าติดตามความเคลื่อนไหวของผู้บุกรุกนั้นเงียบ ๆ เขาได้เฝ้าสังเกตการณ์และบันทึกจุดอ่อนที่เขาพบ ในช่วงเวลา 10 เดือน Stoll พบว่าผู้บุกรุกรายนี้ได้พยายามเจาะเครื่องคอมพิวเตอร์กว่า 400 ระบบทั่วโลก และทำสำเร็จประมาณ 30 ระบบทั้งในมหาวิทยาลัย คอมพิวเตอร์ทางการทหาร และที่อื่น ๆ ด้วย

ผู้บุกรุกรายนี้ไม่เพียงแต่เข้าไปในระบบแล้วสำรวจดูเรื่อยเปื่อยเหมือนที่นักทะลวงทั้งหลายในอินเตอร์เน็ตนิยมทำกัน แต่เขาได้พยายามดูแฟ้มข้อมูลที่น่าสนใจมากอย่างเช่น ข้อมูลเกี่ยวกับ Space Shuttle และเรื่องราวทางการทหารอื่น ๆ

ภายหลังการเฝ้าติดตามพฤติกรรมของผู้บุกรุกรายนี้มา 1 ปีโดยทั้ง FBI และ CIA ก็ได้มีการจับกุมผู้ต้องหาระดับข้ามประเทศ ข้อหาก็เช่นการขายข้อมูลหรือการเป็นสายลับ ถูกทั้งปรับและจำคุกและตัดสิทธิ์ในการออกคะแนนเสียงเลือกตั้ง

นอกจาก 2 กรณีที่กล่าวมาแล้ว ยังมีปัญหาเกี่ยวกับระบบรักษาความปลอดภัยของข้อมูลในคอมพิวเตอร์อีกหลายเรื่อง เช่น การที่เครือข่าย SPAN ของ NASA โดนบุกรุก เครือข่ายของธนาคารของสหรัฐโดนบุกรุก ไวรัสที่ทำให้เครือข่าย MILNET ของทหารต้องแยกออกจากเครือข่ายอื่น ๆ ชั่วคราว และสารพัดไวรัส WORM และ Trojan Horse ที่ติดกันทั่ว

ระบบรักษาความปลอดภัยของระบบปฏิบัติการยูนิกซ์ อาจแบ่งออกเป็น 3 หมวดใหญ่ ๆ คือเรื่องของ account ซึ่งเกี่ยวกับการ login เข้าใช้ระบบ เรื่องของเครือข่าย และเรื่องของระบบไฟล์ 2 เรื่องแรกเป็นเรื่องเกี่ยวกับการอนุญาตให้ผู้ใดเข้าใช้ระบบได้บ้าง ส่วนเรื่องระบบไฟล์นั้นเกี่ยวกับว่าให้ผู้ใช้ใช้ทรัพยากรต่าง ๆ ของระบบโดยเฉพาะแฟ้มข้อมูลต่าง ๆ ได้มากน้อยแค่ไหน และอย่างไร

ความปลอดภัยในระบบ

account

การเจาะระบบที่ง่ายที่สุดทางหนึ่งก็คือการเข้าทางระบบ account นี้เอง เนื่องจากหลายเครื่องมักจะเหลือ account ของผู้ที่ไม่ใช้แล้วในองค์กรนั้น ๆ หรือผู้ใช้มักตั้ง password หรือรหัสที่ง่ายต่อการเดา และอื่น ๆ

PASSWORD

เป็นสิ่งที่สำคัญมากในเรื่องของความปลอดภัยในระบบยูนิกซ์ ถ้าใครสามารถรู้ password ของผู้ใช้คนไหน

เขาก็สามารถทำทุกสิ่งทุกอย่างที่ผู้นั้นทำได้ เช่น ดูหรือแก้ไขข้อมูลส่วนตัวหรือความลับของผู้นั้นได้ และถ้าผู้ใช้คนนั้นเป็น root หรือ superuser ของเครื่องนั้น ปัญหาใหญ่ก็เกิดขึ้นได้แน่ เพราะเขาจะสามารถอ่าน-เขียนไฟล์ทุกไฟล์ในระบบได้เลย ดังนั้นการเลือก password จึงเป็นสิ่งสำคัญมากจริง ๆ

ได้มีการทดลองเกี่ยวกับนิสัยการตั้ง password จากการสุ่มตัวอย่าง 3,289 password มา พบว่า 16 เปอร์เซ็นต์ มีความยาวสั้นกว่า 4 ตัวอักษร และมากถึง 86 เปอร์เซ็นต์ ที่ถือว่าไม่ปลอดภัย 8-30 เปอร์เซ็นต์จะเป็นชื่อ login นั่นเอง หรือกลับจากหน้าไปหลังหรือหลังมาหน้า หรือเอาทั้ง 2 อย่างมาต่อกัน จากการทดลองเอาชื่อผู้หญิงที่นิยมสูงสุด 20 ชื่อตามด้วยตัวเลข 1 ตัวรวมเป็น 200 ชื่อ พบว่าอย่างน้อยจะต้องพบเป็น password อันหนึ่งในแต่ละเครื่องของเครื่องนับร้อย ๆ ที่ทำการสำรวจ

การเลือก PASSWORD

เป้าหมายคือต้องเลือกให้ผู้อื่นเดายากที่สุดที่จะทำได้ เหลือโอกาสให้เขาเพียงแต่การไล่ทดลองทุก ๆ คำจากอักษรตัวเลขและเครื่องหมายทั้งหมดที่เป็นไปได้ ซึ่งการไล่นี้แม้แต่เครื่องที่สามารถไล่ได้เร็วถึง 1 ล้านคำต่อวินาที (ปกติแล้วจะได้อยู่ในระดับร้อยคำต่อวินาทีก็ถือว่าเร็วแล้ว) ก็จะต้องใช้เวลาโดยเฉลี่ยมากกว่า 100 ปีถึงจะไล่ได้ครบ รับรองได้ว่าการไล่ทุกคำอย่างนี้ไม่ได้ผลแน่

จากเป้าหมายที่ว่ามาก็มีหลักการเลือก password คือ

- อย่าใช้ชื่อ login ไม่ว่าจะโดยตรง กลับหน้าหลัง ตัวเล็กบ้างใหญ่บ้าง หรือซ้อนกัน 2 ชุด

- อย่าใช้ชื่อเล่น ชื่อจริง นามสกุล หรือชื่อแฟนหรือลูก ไม่ว่าจะสลับกันอย่างใดก็ตามหรือใช้ตัวเล็กผสมกับใหญ่

- อย่าใช้คำที่มีความหมายเฉพาะสำหรับคุณ เช่น เลขทะเบียนรถ เบอร์โทรศัพท์ ยี่ห้อรถ ชื่อถนนที่บ้าน

- อย่าใช้ตัวอักษรหรือเลขซ้ำกันทั้งหมด เช่น 999999

- อย่าใช้คำที่มีในพจนานุกรมใด ๆ ก็ตามเท่าที่รู้จัก

- ใช้ password ที่ยาวกว่า 6 ตัวอักษร

- ใช้คำที่ผสมตัวเล็กบ้างใหญ่บ้าง

- ใช้ตัวอักษรผสมกับตัวเลขบ้างและกับเครื่องหมายบ้าง

- ใช้ password ที่จำง่ายจะได้ไม่ต้องบันทึกหรือจดไว้

- จะต้องป้อนผ่านคีย์บอร์ดได้อย่างรวดเร็ว เพื่อกันไม่ให้ใครมาแอบดูเวลาป้อน

ถึงกฎเกณฑ์จะมากหน่อยแต่ก็มีวิธีเลือกที่ทำให้จำง่าย อย่างเช่น

- เลือกตัวอักษรจากทุกคำในโครงกลอนหรือสำนวนบางบทเช่น "Protection is not a principle, but an expedient" จะใช้ว่า Pinap, bae เป็นต้น

- สลับบางตัวอักษรหรือตัวสระทำให้เกิดคำที่ไม่มีความหมาย อ่านไม่ออกและทำให้จำง่าย เช่น RoutBoo หรือ QuadpoP

ใช้คำสั้น ๆ 2 คำมาต่อกันด้วยสัญลักษณ์พิเศษ เช่น Book+Mad หรือ OH?rain เป็นต้น

ต้องระดับนโยบายด้วย

นอกจากการขอให้ผู้ใช้เลือก password ที่ดีแล้ว ก็ควรจะมีนโยบายที่ให้ผู้ใช้ต้องปฏิบัติตาม เช่น อย่าเขียน password ลงในกระดาษหรือในแฟ้มข้อมูลใด ๆ ทั้งสิ้น เพราะการระวังรักษากระดาษหรือแฟ้มข้อมูลอย่างนั้น ไม่มีทางสู้ความปลอดภัยจากการเข้ารหัสของเครื่องได้เลย

ห้ามให้ผู้ใช้บอก password แก่ใครทั้งสิ้น เพราะการก็อบปี้แฟ้มข้อมูลให้กัน บางทีชอบใช้วิธีบอก password ให้กันแทนที่จะเซต PERMISSION ของแฟ้มข้อมูลนั้น การบอก password แก่ใครก็ตาม ก็อาจไม่แน่ใจได้ว่าเขาจะไม่บันทึกไว้หรือบอกต่อแก่คนอื่น

และที่สำคัญอีกอย่างก็คือ จะต้องให้มีการเปลี่ยน password ใหม่อย่างน้อยก็ทุก 6 เดือน โดยทั่วไปแล้วระบบการบังคับให้เปลี่ยน password นี้ อาจจะทำได้โดยการใช้ซอฟต์แวร์เพิ่มเติมหรือส่ง memo ไปบอกเพราะระบบวันหมดอายุของ password (password-expiration) มักจะไม่ใช้กัน

นโยบายและหลักการทั้งหมดนี้ควรจะบอกแก่ผู้ใช้ทุกคนในระบบให้ปฏิบัติตาม โดยเฉพาะอย่างยิ่งตัว superuser เอง

วันหมดอายุ

หมายถึงช่วงเวลาที่ผู้ใช้หนึ่ง ๆ ไม่ได้เข้าใช้ account นั้น ๆ ถ้าเจ้าของลาออกไปแล้วหรือจะไม่ใช้เครื่องอีกต่อไปก็ควรจะลบ account นั้นออกจากไฟล์ password เพราะอาจจะมีใครกำลังเจาะเข้าทาง account นี้

ถ้าหากว่าเจ้าของนั้นไปพักร้อนหรือไปเที่ยวต่างประเทศนาน ๆ เป็นสัปดาห์หรือเป็นเดือนก็ควรจะปิดไม่ให้เข้า account นั้นชั่วคราว แต่ไม่ใช่การลบทิ้งเลยทีเดียว โดยการใส่เครื่องหมาย asterisk (*) ลงช่อง password ไว้ ทำให้ไม่สามารถ login เข้าได้ จนกว่าเจ้าของจะกลับมาแล้วมาเปิดใช้ใหม่

GUEST

ปกติแล้ว account นี้จะมีไว้ให้ผู้ที่ใช้เครื่องชั่วคราวในฐานะเป็นแขกหรืออะไรทำนองนั้น ส่วนนี้ก็เป็นช่องโหว่ได้เหมือนกัน ทางที่ดีก็คือลบออกทันทีที่ไม่ต้องการใช้ และต้องไม่ใช้ password ที่ง่าย ๆ อย่าง guest หรือ visitor

ACCOUNT ที่ไม่มี PASSWORD

บางเครื่องจะมี account ที่ไม่มี password เช่น who, date หรือ lpq ไว้เพื่อใช้รันคำสั่งง่าย ๆ โดยไม่ต้อง login เข้าระบบ โดยทั่วไปแล้วเพื่อความสะดวก account เหล่านี้จะไม่ใส่ password และมักจะมี User ID (UID) เป็น 0 หรือ superuser นั่นเอง

และจุดนี้เองที่เป็นช่องโหว่ ถ้าใครสามารถแทนโปรแกรมเหล่านี้ด้วยโปรแกรมของเขาเองเขาก็สามารถใช้ account เหล่านี้รันโปรแกรมในฐานะของ superuser ได้เลยทีเดียว ทางที่ดีที่สุดก็คือให้ทุก account มี password หรือไม่ต้องมี account ประเภทนี้

YELLOW PAGE

ระบบ yellow page ของเครื่อง SUN มีไว้เพื่อให้เครื่องที่อยู่ในเครือข่ายสามารถใช้ไฟล์ password, group และไฟล์อื่น ๆ ร่วมกันได้ โดยไฟล์เหล่านี้จะอยู่ในเครื่องเพียงเครื่องเดียว หลักการก็คือ จะมีบรรทัดพิเศษในไฟล์ password หรือ group ที่จะขึ้นต้นด้วยเครื่องหมาย + อย่างเช่น +::0:0::: ในไฟล์ password หรือ +: ในไฟล์ group เมื่อโปรแกรมอ่านพบบรรทัดเหล่านี้ก็จะไปหาสิ่งที่ต้องการจากเครื่องที่เป็นเซอร์ฟเวอร์ในเครือข่ายนั้นแทนที่จะหาในเครื่องตัวเอง ดังนั้นเครื่องที่เป็นเซอร์ฟเวอร์ก็ไม่จำเป็นต้องมีบรรทัดอย่างนี้

ปัญหาก็อยู่ที่ว่าถ้าหากเครื่องหมายนี้ถูกเผลอลบทิ้งไป บรรทัดนี้ก็จะกลายเป็น account อิสระที่ไม่มี password และก็เป็น superuser ด้วย ซึ่งเป็นการเปิดช่องโหว่ที่อันตรายขนาดใหญ่ให้บรรดาแฮ็กเกอร์หรือผู้ที่ต้องการเจาะระบบ

ความปลอดภัยในเครือข่าย

สมัยนี้ถือเป็นเรื่องปกติมากที่คอมพิวเตอร์ที่ใช้อยู่จะต่อเข้ากับระบบเครือข่ายในท้องถิ่น และในที่สุดก็หมายถึงการต่อเข้าไปในอินเตอร์เน็ตซึ่งเชื่อมโยงเครือข่ายย่อย ๆ ทั่วโลกเข้าหากัน นั่นหมายถึงการป้องกันอันตรายจากผู้ไม่ประสงค์ดี ไม่จำกัดอยู่เพียงแค่ผู้ที่อยู่ในอาคารเท่านั้น แต่ยังหมายถึงใครก็ไม่รู้จากต่างแดนอีกด้วย

ไฟล์ host.equiv

ไฟล์ /etc/host.equiv นี้ใช้บอกรายชื่อเครื่องที่ไว้ใจได้ ซึ่งจะทำให้การรันคำสั่งหรือการ login ข้ามเครื่องด้วยชื่อ login เดียวกัน ทำได้โดยไม่ต้องใส่ password ซึ่งให้ความสะดวกอย่างมาก หรือที่เรียกกันว่า trusted host และนี่ก็เป็นช่องโหว่ที่ต้องระวังเป็นพิเศษทีเดียว เพื่อความปลอดภัยแล้วอย่าใส่ชื่อลักษณะของ wildcard เข้าไป ให้เขียนชื่อเครื่องที่ถูกต้องเข้าไปเต็ม ๆ

ไฟล์ .rhosts

หลักการคล้าย ๆ กับไฟล์ host.equiv เพียงแต่เป็นการให้สิทธิ์หรือความไว้วางใจในระดับผู้ใช้แต่ละคนเท่านั้น หมายถึงผู้ใช้ที่มีไฟล์นี้ที่บ้านตัวเองจะยอมให้ account ชื่อเดียวกันในเครื่องอื่นสามารถเข้าไปใช้ได้โดยที่อาจไม่เป็นเครื่อง trusted host กัน อันตรายดูเหมือนจะมากกว่า เนื่องจากว่าเป็นการเซตโดยผู้ใช้ธรรมดาทั่ว ๆ ไป ซึ่งอาจไม่ใส่ใจนักกับเรื่องของความปลอดภัยเท่ากับ superuser ซึ่งเซต host.equiv ได้ผู้เดียวเท่านั้น

การป้องกันอันตรายที่อาจเกิดขึ้นดีที่สุดก็คือปิด ไม่อนุญาตให้มีการใช้ไฟล์นี้ในระบบ

SECURE TERMINAL

ในยูนิกซ์รุ่นใหม่ ๆ จะมีหลักการนี้โดยการใส่คำว่า secure ไว้ในไฟล์ /etc/ttytab เพื่อบอกว่าเทอร์มินัลนั้นๆ สามารถ login root ได้ และเทอร์มินัลที่ไม่มีจะไม่สามารถ login root ได้ (แต่สามารถใช้คำสั่ง su เพื่อเป็น superuser ได้) ตัวอย่างเช่นในรูปที่ 1 (หากมีการแก้ไขไฟล์จะต้องบอก process

```
console   "/usr/etc/getty std.9600"  sun      off secure
ttya      "/usr/etc/getty std.9600"  unknown  off secure
ttyb      "/usr/etc/getty std.9600"  unknown  off secure
ttyp0     none                       network  off secure
ttyp1     none                       network  off secure
ttyp2     none                       network  off secure
```

รูปที่ 1
ตัวอย่างไฟล์ /etc/ttytab
ซึ่งมีการตั้งเป็น secure terminal

```
/usr
/home
/var/spool/mail
#
/export/root/client1    -access=client1,root=client1
/export/swap/client1    -access=client1,root=client1
#
/export/root/client2    -access=client2,root=client2
/export/swap/client2    -access=client2,root=client2
```

รูปที่ 2
ตัวอย่างของไฟล์ /etc/exports

init ให้อ่านไฟล์นี้ใหม่ด้วยคำสั่ง #kill -HUP

ปกติแล้วเครื่อง SUN จะตั้งเทอร์มินัลทุกอันในไฟล์นี้เป็น secure รวมทั้ง psuedo เทอร์มินัลด้วย ซึ่งใช้ในการ login ข้ามเครื่องภายในเครือข่าย หากจะให้ปลอดภัยแล้วควรจะตั้ง secure ไว้เฉพาะเทอร์มินัลที่ต่อโดยตรงหรือเพียงแค่จอคอลโซลได้ดี

แล้วถ้าต้องการให้ปลอดภัยจริง ๆ แล้วควรจะปิดไม่ให้มีการตั้ง secure เลย โดยหากต้องการเป็น root หรือ superuser จะต้อง login ชื่อตัวเองก่อน แล้วจึงใช้คำสั่ง su เพื่อแปลงร่างเท่านั้น ทั้งนี้จะได้รู้ว่าใครมาใช้อำนาจของ superuser

Network File System

หรือ NFS ได้ถูกสร้างขึ้นมาเพื่อให้เกิดการใช้ไฟล์ร่วมกันภายในระบบเครือข่าย โดยทั่วไปก็คือให้เวอร์กสเตชันไม่มีดิสก์ซึ่งตั้งในสำนักงานไปใช้ดิสก์ของคอมพิวเตอร์ขนาดใหญ่ในศูนย์คอมพิวเตอร์ได้

ไฟล์ exports

ไฟล์ /etc/exports นี้จะบอกว่าระบบไฟล์ (FILE SYSTEM) ไหนให้เครื่องอื่นใช้โดยคำสั่ง mount ได้บ้างคล้าย ๆ กับการส่งออกนั่นเอง ตัวอย่างของไฟล์นี้ได้แก่ในรูปที่ 2 คำว่า root= จะบอกรายชื่อเครื่องที่อนุญาตให้ใช้ระบบไฟล์ในบรรทัดนั้นแบบ superuser ได้ ส่วนคำว่า access= จะบอกรายชื่อเครื่อง (แยกแต่ละชื่อด้วย colon) ที่อนุญาตให้ใช้ระบบไฟล์ได้ ถ้าไม่มี access= จะหมายถึงทุกเครื่องในระบบเครือข่ายสามารถใช้ระบบไฟล์นั้นได้ ดังนั้นจะต้องรอบคอบในการกำหนดชื่อเครื่องลงในแต่ละบรรทัดของไฟล์ /etc/exports นี้ และเมื่อมีการแก้ไขไฟล์นี้ก็จะต้องเรียกคำสั่ง #exportfs -a เพื่อใช้งานของใหม่ที่แก้ไขแล้ว

ไฟล์ netgroup

เนื่องจากแต่ละบรรทัดของไฟล์ /etc/exports ถูกจำกัดให้เขียนชื่อเครื่องได้เพียง 10 ดังนั้นจึงมีหลักการของกลุ่มทางเครือข่ายมาเพื่อลดข้อจำกัด

```
A_Group       (servera,,) (clienta1,,) (clienta2,,)

B_Group       (serverb,,) (clientb1,,) (clientb2,,)

AdminStaff    (clienta1,mary,) (clientb3,joan,)

AllSuns       A_Group B_Group
```

รูปที่ 3 ตัวอย่างไฟล์ /etc/netgroup

```
/usr                        -access=A_Group
/home                       -access=A_Group:B_Group
/var/spool/mail             -access=AllSuns
#
/export/root/client1        -access=client1,root=client1
/export/swap/client1        -access=client1,root=client1
#
/export/root/client2        -access=client2,root=client2
/export/swap/client2        -access=client2,root=client2
```

รูปที่ 4 ตัวอย่างไฟล์ /etc/exports ที่ใช้ netgroup

นั้นลง โดยบอกชื่อกลุ่มและสมาชิกในไฟล์ /etc/netgroup

จากตัวอย่างในรูปที่ 3 จะบอกว่ามีการแบ่งกลุ่มทางเครือข่ายออกเป็น 4 กลุ่มคือ A__Group, B__Group, AdminStaff และ AllSuns กลุ่ม AllSuns เป็นการรวมกลุ่มย่อย 2 กลุ่ม คือ A__Group และ B__Group เข้าด้วยกัน

ปกติการกำหนดชื่อกลุ่มนี้จะเป็น 3 ส่วนคือ (host, user, domain) โดยทั่วไปแล้วช่อง domain จะไม่ได้ใช้และปล่อยว่างไป ถ้าช่อง host หรือ user ถูกปล่อยว่างจะหมายถึงทุก host หรือทุก user เช่น (host__a,,) หมายถึงทุก user ในเครื่อง host__a และ (,user__b,) จะหมายถึงผู้ใช้ชื่อ user__b ในทุก ๆ เครื่อง

รูปที่ 4 เป็นการเพิ่มเติมจากในรูปที่ 2 ระบบไฟล์ /usr ตอนนี้สามารถใช้ได้โดยสมาชิกใน A__Group ได้แก่ ทุก user ในเครื่อง servera, clienta 1 และ clienta2 เพื่อความปลอดภัยจึงต้องระมัดระวังในการกำหนดและสร้างระบบกลุ่มทางเครือข่ายเหล่านี้ให้ดี

Superuser access

โดยทั่วไปแล้ว NFS จะเปลี่ยน UID ของ root ในเครื่องอื่นที่เข้ามาใช้ให้เป็น nobody เพื่อความปลอดภัยไม่ให้ root จากเครื่องอื่นเข้ามากลายเป็น root ในเครื่องเราได้ตรง ๆ เพื่อความสะดวกแล้ว ถ้าต้องการให้เป็น root ในเครื่องเราได้ก็จะต้องใส่ root= ลงไป อย่างในรูปที่ 2 และ 4 ดังนั้นการใส่ root= ลงไปเท่ากับเป็นการเปิดอกรับ root จากเครื่องที่เราใส่ชื่อไว้อย่างเต็มที่ จึงต้องระมัดระวังชื่อที่ใส่นี้ให้รอบคอบ

FTP

โปรแกรม ftp และ ftpd มีไว้ให้ผู้ใช้สามารถติดต่อกับเครื่องภายในเครือข่ายและรับส่งแฟ้มข้อมูลกันได้ ปัญหาก็อยู่ที่ว่าเวอร์ชันเก่า ๆ ของโปรแกรมนี้มีบักอยู่มาก และเป็นช่องว่างที่ทำให้นักเจาะทั้งหลายเล่นงานเครื่องได้ ถ้าเครื่องที่ใช้มี ftpd ที่เก่ากว่า ธ.ค. 1988 ควรจะหาเวอร์ชันที่ใหม่กว่าซึ่งปลอดภัยกว่ามาใช้

ftp จะอนุญาตให้ผู้ใช้ที่ไม่มี account ในเครื่องสามารถเข้ามาดึงแฟ้มข้อมูลหรือให้แฟ้มข้อมูลได้โดยใช้ login ชื่อ anonymous โดยจะอนุญาตให้ทำงานได้กับไดเรคทอรีเฉพาะบริเวณหนึ่งเท่านั้น เพื่อความปลอดภัยคุณควรจะ

- สร้าง account 'ftp', ใส่ช่อง password เป็น * และสร้างบ้านพิเศษไว้ให้ เช่น /usr/ftp หรือ /usr/spool/ftp

```
# chown ftp ~ftp
# chmod 555 ~ftp
```

รูปที่ 5

```
# mkdir ~ftp/bin
# chown root ~ftp/bin
# chmod 555 ~ftp/bin
# cp -p /bin/ls ~ftp/bin
# chmod 111 ~ftp/bin/ls
```

รูปที่ 5.1

```
# mkdir ~ftp/etc
# chown root ~ftp/etc
# chmod 555 ~ftp/etc
# cp -p /etc/passwd /etc/group ~ftp/etc
# chmod 444 ~ftp/etc/passwd ~ftp/etc/group
```

รูปที่ 5.2

```
# mkdir ~ftp/pub
# chown ftp ~ftp/pub
# chmod 777 ~ftp/pub
```

รูปที่ 5.3

```
% tftp
tftp> connect yourhost
tftp> get /etc/motd tmp
Error code 1: File not found
tftp> quit
%
```

รูปที่ 6
วิธีการตรวจสอบ TFTP

- เปลี่ยนโหมดไดเรคทอรีนั้นให้ทุกคนเขียนได้ ตามรูปที่ 5

- สร้างไดเรคทอรี ~ftp/bin เป็นของ superuser และไม่ให้ผู้อื่นเขียนได้ แล้วก็อบปี้โปรแกรม ls มาไว้ที่นี่ตามรูปที่ 5.1

- สร้างไดเรคทอรี ~ftp/etc เป็นของ superuser และไม่ให้ผู้อื่นเขียนได้ แล้วก็อบปี้ไฟล์ password กับ group มาที่นี่ เปลี่ยนช่อง password เป็น * ให้หมด อาจจะลบ account อื่นทั้งหมด แต่ยกเว้นจะต้องเก็บ ftp ไว้ดังในรูปที่ 5.2

- สร้างไดเรคทอรี ~ftp/pub เป็นของ ftp และให้ทุกคนเขียนได้ ซึ่งจะเป็นที่ ๆ ให้ทุกคนก็อบปี้ไฟล์เข้าไปให้ anonymous FTP ใช้งานได้ดังในรูปที่ 5.3

TRIVIAL FTP

TFTP เป็นโปรแกรมที่คล้าย ๆ กับ FTP ขนาดย่อ ไม่มีการตรวจสอบผู้เข้าใช้ และการติดต่อจะใช้โปรโตคอล UDP (User Datagram Protocol) แทน TCP (Transmission Control Protocol) เนื่องจากเป็นคล้ายกับการย่อส่วนลงมา จึงทำให้ TFTP หลายตัวมีช่องโหว่ด้านความปลอดภัย ตรวจสอบได้ในรูปที่ 6 ถ้า TFTP ที่ใช้ไม่ตอบว่า "File not found" แต่กลับมีการส่งไฟล์ คุณควรจะเปลี่ยนตัว tftpd ให้เป็นเวอร์ชันใหม่ขึ้น (เวอร์ชันของ SunOs รุ่นก่อนรีลิส 4.0 จะพบปัญหานี้)

MAIL

โดยทั่วไปอีเมล์ (Electronic Mail) หรือจดหมายทางอิเล็กทรอนิกส์นับว่าเป็นวัตถุประสงค์อันหนึ่งที่ต่อคอมพิวเตอร์เราเข้าไว้ในระบบเครือข่าย ปัญหาก็คล้าย ๆ กับ FTP คือในเวอร์ชันเก่า ๆ จะพบว่ามีบักอยู่ซึ่งทำให้เกิดปัญหาด้านความปลอดภัยได้ ซึ่งโปรแกรมลักษณะ WORM ในอินเตอร์เน็ตได้ใช้ช่องว่างนี้สำเร็จแล้ว ปัจจุบันพบว่าโปรแกรม sendmail ตัวปัญหานี้ได้แก้ช่องว่างเหล่านั้นไปแล้ว อย่างไรก็ตาม

มีข้อควรระวังในการใช้ระบบอีเมล์ดังต่อไปนี้

- ลบ alias 'decode' ออกจากไฟล์ aliases (/etc/aliases หรือ /usr/lib/al สามารถเอาเชลล์หรือส่งข้อความไปให้เชลล์ได้ด้วยโปรแกรมเหล่านั้น

- ตรวจดูอย่าให้โปรแกรม sendmail ใช้คำสั่ง debug ได้ โดยตรวจสอบดังในรูปที่ 7 ถ้าตอบด้วย '200 Debug set' หมายถึงว่ามีช่องโหว่ให้ถูกโจมตีได้ ควรจะหา sendmail เวอร์ชันที่ใหม่กว่ามาแทน

FINGER

โปรแกรม finger ช่วยให้เราสามารถดูข้อมูลบางอย่างของผู้ใช้เครื่องได้ เช่น ชื่อเต็ม, home directory, เวลา login ครั้งสุดท้าย หรือในบางครั้งบอกด้วยว่าได้รับจดหมายหรืออ่านจดหมายครั้งสุดท้ายเมื่อใด และโปรแกรม fingerd ทำให้สามารถดูข้อมูลเหล่านี้ของผู้ใช้ในเครื่องอื่นที่อยู่ในเครือข่ายเดียวกันได้

โปรแกรม fingerd รุ่นเก่า ๆ มีช่องโหว่ที่ WORM ใช้เจาะระบบมาแล้ว หากเวอร์ชันที่คุณใช้อยู่เก่ากว่า 6 พ.ย. 1988 ก็ควรจะหาเวอร์ชันที่ใหม่กว่ามาแทนได้แล้ว

โมเด็มและเทอร์มินัลเซอร์ฟเวอร์

เป็นอีกทางหนึ่งที่พวกแฮ็กเกอร์ใช้กันด้วย หากเซตไม่ดีมันอาจจะกลายเป็นทางที่ถูกใช้เจาะ ควรจะตรวจสอบดูโดย

- ถ้าผู้ใช้โทรโมเด็มเข้ามา login ในระบบแล้วสายนั้นหลุดไป ระบบจะต้อง logout ผู้ใช้คนนั้นออก ถ้าไม่ให้ตรวจดูการต่อฮาร์ดแวร์และ configuration ของ kernel ของพอร์ตนั้น

- เมื่อผู้ใช้ logout การทำงานระบบควรจะให้สายโมเด็มนั้นหลุดหรือขาดไป (hang up) ถ้าไม่ก็ต้องตรวจสอบอย่างข้อที่แล้ว

- ถ้าการติดต่อจาก Terminal Server ขาดไป ระบบควรจะ logout ผู้ใช้คนนั้นออก

- ถ้าเทอร์มินัลเซอร์ฟเวอร์ถูกต่ออยู่กับโมเด็มและผู้ใช้วางสายนั้นขาดไป เทอร์มินัลเซอร์ฟเวอร์ควรจะบอกระบบกลางว่าผู้ใช้ทำสายขาดไปแล้ว

FIREWALL

เป็นหลักการที่ให้เครื่อง host ตัวหนึ่งเป็นตัวเชื่อมระหว่างเครือข่ายภายในกับเครือข่ายภายนอกเป็นเครื่อง ๆ เดียวที่เครือข่ายภายนอกเห็นเครือข่ายภายในคล้ายกับเป็นเครื่องตัวแทน เพื่อความปลอดภัยควรจะ

- ผู้ใช้ในเครือข่ายข้างในจะต้อง login เข้าเครื่องที่เป็น FIREWALL นี้ก่อนถึงจะติดต่อภายนอกได้ และเช่นกันการจะเข้ามาในเครือข่ายภายในก็ต้องผ่านการ login เข้าเครื่องที่เป็นประตูหรือ FIREWALL นี้ก่อนเพื่อความปลอดภัยถึงจะไม่สะดวกก็ตาม

- อีเมล์ที่จะส่งออกจะต้องส่งผ่านไปที่ FIREWALL ก่อนจึงจะออกข้างนอกได้ และอีเมล์จากภายนอกก็จะเข้ามาโดยผ่าน FIREWALL นี้ก่อนด้วย ทั้งนี้ทำได้โดยอาจจะใช้กลไกของการ aliases หรือใช้ name server MX record

- เครื่องที่เป็น FIREWALL นี้จะต้องไม่ใช้ระบบไฟล์ของ NFS และก็ต้องไม่ให้เครื่องอื่นใส่ (mount) ระบบไฟล์ของตนไปใช้ด้วย

- ระบบ password จะต้องปฏิบัติใช้อย่างเข้มงวด

- เครื่องนี้จะต้องไม่มีการใช้ trusted host หรือเป็น trusted host ของเครื่องอื่น

สรุปก็คือต้องรักษาความปลอดภัยเครื่องนี้อย่างดีที่สุด ถ้าเครื่องนี้ถูกเจาะได้แล้วเครื่องอื่น ๆ ในเครือข่ายภายในก็จะถูกคุกคามได้

```
% telnet localhost 25
220 yourhost Sendmail 5.61 ready at 9 Mar 90 10:57:36 PST
debug
500 Command unrecognized
quit
%
```

รูปที่ 7 วิธีการตรวจสอบโปรแกรม sendmail

ความปลอดภัยของระบบไฟล์

ด่านสุดท้ายที่จะรักษาความปลอดภัยแก่ระบบก็คือบิตต่าง ๆ ที่บอกว่าแต่ละไฟล์หรือไดเรคทอรีอนุญาตให้เจ้าของ ผู้ที่อยู่ในกลุ่ม และผู้อื่นใช้งานได้อย่างไรบ้าง เรียกกลุ่มบิตเหล่านี้ว่า OWNER PERMISSION, GROUP PERMISSION และ WORLD PERMISSION ตามลำดับ แต่ละอย่างจะประกอบด้วยชุดของบิตดังนี้

READ ถ้าเป็น 1 บอกว่าอ่านไฟล์หรือไดเรคทอรีนั้นได้ สำหรับไดเรคทอรีจะหมายถึงว่าดูส่วนประกอบคือชื่อของไฟล์ภายในนั้นได้ (แต่ไม่ได้หมายความว่าจะเข้าถึงไฟล์เหล่านั้นได้)

WRITE ถ้าเป็น 1 บอกว่าแก้ไขไฟล์หรือไดเรคทอรีนั้นได้ สำหรับไดเรคทอรีจะหมายถึงจะสามารถสร้าง ลบ หรือเปลี่ยนชื่อไฟล์ภายใต้นั้นได้ การที่จะลบไฟล์ ๆ หนึ่งได้หรือไม่นั้นไม่ได้ขึ้นอยู่กับ PERMISSION ของไฟล์นั้น แต่ขึ้นอยู่กับ PERMISSION ของไดเรคทอรีที่ไฟล์นั้นอยู่ต่างหาก

EXECUTE ถ้าเป็น 1 หมายถึงสามารถรันหรือค้นหาไฟล์หรือไดเรคทอรีนั้นได้ ถ้าเป็นไดเรคทอรีจะหมายถึงสามารถเข้าถึงไฟล์ภายในนั้นได้

นอกจากนี้ยังมีบิตพิเศษคือบิตที่ 4 ซึ่งไม่มีผลกับไฟล์ที่รันไม่ได้ มีความหมายต่างกันคือ

SETUID ถ้าเซตใน OWNER PERMISSION บิตนี้จะหมายถึงเมื่อโปรแกรมนั้นถูกรันไม่ว่าโดยใคร USER ID ขณะนั้นจะเป็นของเจ้าของไฟล์หรือโปรแกรมนั้น อย่างเช่น โปรแกรม sendmail ซึ่งเป็นของ root จะเซต SETUID BIT ไว้ ซึ่งทำให้สามารถเขียนไฟล์ในบริเวณที่เก็บจดหมาย ซึ่งปกติแล้วผู้ใช้ทั่ว ๆ ไปจะไม่สามารถทำได้

SETGID ถ้าเซตใน GROUP PERMISSION ลักษณะจะคล้ายกับ SETUID ต่างกันตรงที่เป็นการเปลี่ยน GROUP ID แทน

STICKY BIT ถ้าเซตใน WORLD PERMISSION จะบอกระบบปฏิบัติการให้ปฏิบัติต่อ TEXT IMAGE ของโปรแกรมพิเศษจากปกติ ในยูนิกซ์รุ่นใหม่ ๆ ถ้าเซต STICKY BIT นี้กับไดเรคทอรี จะหมายถึงว่าไม่อนุญาตให้ลบหรือเปลี่ยนชื่อไฟล์ของผู้อื่นในไดเรคทอรีนั้น มีประโยชน์โดยตรงกับไดเรคทอรี /tmp นั่นเอง วิธีการเซตคือ

```
#chmod o+t directory
```

เพื่อความปลอดภัยแล้วไม่ควรมีเชลล์สคริปต์ที่เซต SETUID BIT หรือ SETGID BIT เนื่องจากเป็นช่องทางที่ทำให้ผู้ที่รันมีความสามารถกลายเป็นเจ้าของหรือกลุ่มเดียวกันได้

UMASK

เป็นบิตที่บอกไม่ให้เซตบิต PERMISSION เป็นหนึ่งสำหรับไฟล์ที่เพิ่งสร้างใหม่ เช่น ถ้าไม่ต้องการให้ไฟล์ใหม่ของเราอ่าน เขียน หรือรันได้โดยผู้อื่นทั้งหมดก็ตั้งค่า UMASK เป็น 077 โดยทั่วไปแล้วจะตั้งค่า UMASK นี้ใน .cshrc หรือ .profile และสำหรับ superuser ควรตั้งค่าเป็น 022 ใน /.cshrc เพื่อไม่ให้ผู้อื่นแก้ไขไฟล์ของ superuser

การ ENCRYPT ไฟล์

คำสั่ง crypt มาตรฐานของยูนิกซ์มีไว้เพื่อเข้ารหัสข้อมูลที่ต้องการปกปิด

แต่ว่าสำหรับผู้ที่ต้องการจะล้วงความลับแล้วมันเป็นเพียงอุปสรรคเล็กน้อยเท่านั้น และปัจจุบันนี้ก็มีโปรแกรมที่ทำขึ้นมาเพื่อไขไฟล์ที่ถูก crypt แล้วด้วย

ในวงการ encryption ดูเหมือนว่า DES หรือ Data Encryption Standard จะได้รับการยอมรับว่าปลอดภัยมากที่สุดวิธีหนึ่ง (DES จัดเป็น symmetric encryption scheme เนื่องจากทั้งการเข้ารหัสและถอดรหัสจะใช้คีย์หรือรหัสเดียวกัน) แต่อย่างไรก็ตามยังไม่มีการพิสูจน์ได้แน่ชัดว่า DES นี้จะปลอดภัย 100 เปอร์เซ็นต์ ทางที่ดีที่สุดก็คืออย่าเก็บข้อมูลที่สำคัญไว้ในคอมพิวเตอร์โดยเฉพาะเครื่องพีซีหรือยูนิกซ์

ส่วนกลไกของไฟล์ /etc/password ซึ่งจะเก็บ password สำหรับการ login ได้ใช้อัลกอริทึมพิเศษที่ดัดแปลงมาจาก DES เพื่อให้เป็นการเข้ารหัสทางเดียวโดยไม่สามารถถอดรหัสกลับได้ ในขณะที่ login เครื่องจะทำการเข้ารหัสสิ่งที่คีย์เข้าไปแล้วนำไปเปรียบเทียบกับค่าที่เข้ารหัสไว้แล้วในไฟล์ /etc/password ไม่ได้ใช้วิธีถอดรหัสค่าในไฟล์ /etc/password ออกมาเปรียบเทียบกับที่คีย์เข้าไป

DEVICE

ไฟล์ต่าง ๆ ที่อยู่ในไดเรคทอรี /dev เป็นตัวแทนในการเข้าถึงอุปกรณ์ทั้งหลายที่พ่วงต่ออยู่กับตัวเครื่อง ตั้งแต่เทอร์มินัลทั้งหลายจนถึงดิสค์และแม้แต่หน่วยความจำ หากไม่มีการป้องกันที่เหมาะสมแล้วไฟล์เหล่านี้จะเป็นประตูต้อนรับพวกแฮ็กเกอร์ให้เข้าเจาะระบบได้ มีข้อควรระวังในการเซตผู้เป็นเจ้าของกลุ่มและ PERMISSION ไฟล์ต่าง ๆ เหล่านี้มากมาย ซึ่งอาจต่างกันไปบ้างเล็กน้อยในเครื่องต่างกัน ในที่นี้จะกล่าวถึงเฉพาะบางส่วนคือ

- ไฟล์ /dev/kmem, /dev/mem และ /dev/drum จะต้องไม่ให้ผู้อื่นอ่านได้ ถ้ามีกลุ่ม kmem และยูทิลิตี้พวก ps มีการเซตบิต SETGID เป็น kmem แล้ว ไฟล์เหล่านี้ควรจะมีเจ้าของเป็น root กลุ่มเป็น kmem และโหมด 640 แต่ถ้าไม่มีกลุ่ม kmem และยูทิลิตี้อย่าง ps มีการเซตบิต SETUID เป็น root แล้ว ควรจะมีเจ้าของเป็น root และมีโหมด 600

อุปกรณ์พวกดิสค์อย่าง /dev/sd0a ควรจะมี root เป็นเจ้าของและกลุ่มเป็น operator และโหมด 640 แต่ละดิสค์เหล่านี้จะมี 8 พาติชัน และแต่ละพาติชันจะมีไฟล์เสมือนจำนวน 2 ไฟล์ใน /dev ดูรูปที่ 8 เป็นรายชื่อไฟล์ที่เกี่ยวข้องกับดิสค์ sd0

- นอกจากนั้นไฟล์เกือบทั้งหมดใน /dev ควรจะมี root เป็นเจ้าของยกเว้นเทอร์มินัลซึ่งจะมีเจ้าของเป็นผู้ที่กำลัง login อยู่ขณะนั้น พอ logout ออกก็จะคืนมาเป็นของ root เหมือนเดิม

มาถึงตอนนี้แล้วก็ได้ทราบวิธีการต่าง ๆ มากมายที่จะป้องกันรักษาความปลอดภัยให้กับระบบยูนิกซ์ แต่ว่าระบบ

```
sd0a    sd0e    rsd0a    rsd0e
sd0b    sd0f    rsd0b    rsd0f
sd0c    sd0g    rsd0c    rsd0g
sd0d    sd0h    rsd0d    rsd0h
```

รูปที่ 8 รายชื่อไฟล์ที่เกี่ยวข้องกับดิสค์ sco ในไดเรคทอรี /dev

ปฏิบัติการยูนิกซ์ที่ลงกันใหม่ ๆ ส่วนมากจะเปิดใช้มาตรการเหล่านี้ไม่ครบ จึงเป็นหน้าที่ของ system administrator ที่จะตรวจสอบและใช้งานมาตรการทั้งหลายเหล่านี้

นอกจากการวางมาตรการป้องกันต่าง ๆ ดังที่ได้กล่าวมาแล้ว การคอยสอดส่องดูแลระบบก็เป็นเรื่องจำเป็นสำหรับการรักษาความปลอดภัยเช่นกัน

การตรวจสอบระบบ

account

ควรจะมีการตรวจสอบการ login และ logout เสมอ ๆ เพื่อตรวจดูว่ามีใคร login ในเวลาที่ผิดปกติหรือเปล่า เช่น เจ้าของลาพักร้อนไปต่างประเทศแล้วใครมาใช้ หรือการใช้คำสั่งที่ปกติใช้ไม่ได้

ไฟล์ LASTLOG

ไฟล์ /use/adm/lastlog จะเก็บวันเวลาที่ผู้ใช้แต่ละคน login เข้าระบบครั้งสุดท้ายข้อความที่แสดงในตอน login เช่น

```
Last login: Wed Jul 9 14:31:57 from
chulkn.chula.ac.th
```

ก็ได้ข้อมูลมาจากไฟล์นี้และรวมทั้งที่รายงานเวลา login ครั้งล่าสุดจากโปรแกรม finger ไฟล์นี้จะช่วยบอกให้รู้อย่างง่าย ๆ ว่ามีใคร login ในเวลาที่ไม่ควรได้

ไฟล์ UTMP และ WTMP

ไฟล์ /etc/utmp เก็บข้อมูลว่ามีใคร login เข้าระบบบ้าง คำสั่ง who ใช้ข้อมูลจากไฟล์นี้ดูได้เช่นในรูปที่ 9

ส่วนไฟล์ /usr/adm/wtmp จะเก็บเวลา login และ logout ของผู้ใช้ทุกคน คำสั่ง who สามารถดูข้อมูลในไฟล์นี้ได้เช่นในรูปที่ 9.1

บรรทัดที่มีชื่อ login name บอกเวลา login ส่วนบรรทัดที่ไม่มีชื่อบอกเวลา logout แต่โดยทั่วไปแล้วผลลัพธ์ที่ได้จะไม่อ่านง่ายแบบนี้ ถ้ามีการ login กับ logout จากผู้ใช้หลาย ๆ คน เวลา login และ logout จะปนเปกันไม่เรียงสวยงาม จะต้องจัดเรียงเองโดยอาศัยชื่อเทอร์มินัล

คำสั่ง last ใช้ดูข้อมูลใน wtmp ได้ด้วยโดย last จะช่วยเรียงตามเวลาและจับคู่เวลา login และ logout ให้อย่างเช่นในรูปที่ 9.2

เวลาปิด-เปิดเครื่องโดยคำสั่ง

```
% who
hendra   tty0c   Mar 13 12:31
heidari  tty14   Mar 13 13:54
welgem   tty36   Mar 13 12:15
reagin   ttyp0   Mar 13 08:54   (aaifs.itstd.sri.)
ghg      ttyp1   Mar  9 07:03   (hydra.riacs.edu)
compion  ttyp2   Mar  1 03:01   (ei.ecn.purdue.ed)
```

รูปที่ 9 ตัวอย่างผลลัพธ์จากโปรแกรม who

```
% who /usr/adm/wtmp
davy     ttyp4    Jan  7 12:42 (annex01.riacs.ed)
         ttyp4    Jan  7 15:33
davy     ttyp4    Jan  7 15:33 (annex01.riacs.ed)
         ttyp4    Jan  7 15:35
hyder    ttyp3    Jan  8 09:07 (triceratops.itst)
         ttyp3    Jan  8 11:43
```

รูปที่ 9.1 ตัวอย่างผลลัพธ์จากโปรแกรม who เพื่อดูข้อมูลในไฟล์ /usr/adm/wtmp

shutdown และ reboot ก็เห็นได้จากคำสั่ง last นี้ได้ด้วย และในยูนิกซ์รุ่นใหม่ ๆ ก็บันทึกการใช้ ftp ในนี้ด้วย (ดังในบรรทัดสุดท้าย)

ไฟล์ ACCT

ไฟล์ /usr/adm/acct บันทึกการใช้คำสั่งต่าง ๆ ว่าใครใช้คำสั่งอะไร เวลาเท่าใด ที่จอเทอร์มินัลไหน โดยจะต้องกำหนด kernel ใช้ออปชัน SYSACCT ไฟล์นี้ดูได้ด้วยคำสั่ง lastcomm ตัวอย่างเช่นในรูปที่ 10

คอลัมน์แรกบอกชื่อคำสั่งถัดไป เป็นแฟล็กเช่น 'F' บอกว่า process มีการสร้าง process ลูก 'S' บอกว่ารันด้วยการ SETUID 'D' บอกว่าสิ้นสุดด้วยการ core dump และ 'x' บอกว่าถูกทำให้จบอย่างไม่ปกติ (abnormally killed) ที่เหลือบอกว่าใครรันที่จอไหน ใช้เวลาซีพียูเท่าใด และเริ่มเมื่อวันเวลาอะไร

```
% last
davy       ttyp3  intrepid.itstd.s Tue Mar 13 10:55 - 10:56 (00:00)
hyder      ttyp3  clyde.itstd.sri. Mon Mar 12 15:31 - 15:36 (00:04)
reboot     ~                       Mon Mar 12 15:16
shutdown   ~                       Mon Mar 12 15:16
arms       ttyp3  clyde0.itstd.sri Mon Mar 12 15:08 - 15:12 (00:04)
hyder      ttyp3  spam.itstd.sri.c Sun Mar 11 21:08 - 21:13 (00:04)
reboot     ~                       Sat Mar 10 20:05
davy       ftp    hydra.riacs.edu  Sat Mar 10 13:23 - 13:30 (00:07)
```

รูปที่ 9.2 ตัวอย่างผลลัพธ์จากโปรแกรม last

การตรวจสอบระบบเครือข่าย

การตรวจสอบความปลอดภัยของเครื่องที่ต่อเข้ากับระบบเครือข่ายยากกว่าเครื่องที่อยู่เดี่ยว ๆ เนื่องจากคล้ายกับเป็นการเปิดตัวต้อนรับแฮ็กเกอร์ทั้งหลาย แต่ว่าก็มีโปรแกรมหลายตัวที่ช่วยได้ เช่น

SYSLOG

เป็นกลไกที่ให้คำสั่งต่าง ๆ ทำการล็อกข้อความของความผิดปกติต่าง ๆ ออกที่จอคอนโซลและลงที่ไฟล์ /usr/adm/messages รวมทั้งวันเวลา ชื่อโปรแกรมและหมายเลข process (PID) เช่นในรูปที่ 11

ที่น่าสนใจก็คือบรรทัดที่ล็อกโดย login และ su ซึ่งช่วยให้เรารู้ว่าใคร

```
% lastcomm
sh          S    root     __          0.67 secs Tue Mar 13 12:45
atrun            root     __          0.23 secs Tue Mar 13 12:45
lpd         F    root     __          1.06 secs Tue Mar 13 12:44
lpr         S    burwell  tty09       1.23 secs Tue Mar 13 12:44
troff            burwell  tty09      12.83 secs Tue Mar 13 12:44
eqn              burwell  tty09       1.44 secs Tue Mar 13 12:44
df               kindred  ttyq7       0.78 secs Tue Mar 13 12:44
ls               kindred  ttyq7       0.28 secs Tue Mar 13 12:44
cat              kindred  ttyq7       0.05 secs Tue Mar 13 12:44
stty             kindred  ttyq7       0.05 secs Tue Mar 13 12:44
tbl              burwell  tty09       1.08 secs Tue Mar 13 12:44
rlogin      S    jones    ttyp3       5.66 secs Tue Mar 13 12:38
rlogin      F    jones    ttyp3       2.53 secs Tue Mar 13 12:41
stty             kindred  ttyq7       0.05 secs Tue Mar 13 12:44
```

รูปที่ 10 การดูข้อมูลในไฟล์ /use/adm/acct โดย /astcomm

```
Mar 12 14:53:37 sparkyfs login: ROOT LOGIN ttyp3 FROM setekfs.itstd.sr
Mar 12 15:18:08 sparkyfs login: ROOT LOGIN ttyp3 FROM setekfs.itstd.sr
Mar 12 16:50:25 sparkyfs login: ROOT LOGIN ttyp4 FROM pongfs.itstd.sri
Mar 12 16:52:20 sparkyfs vmunix: sd2c:  read failed, no retries
Mar 13 06:01:18 sparkyfs vmunix: /: file system full
Mar 13 08:02:03 sparkyfs login: ROOT LOGIN ttyp4 FROM triceratops.itst
Mar 13 08:28:52 sparkyfs su: davy on /dev/ttyp3
Mar 13 08:38:03 sparkyfs login: ROOT LOGIN ttyp4 FROM triceratops.itst
Mar 13 10:56:54 sparkyfs automount[154]: host aaifs not responding
Mar 13 11:30:42 sparkyfs login: REPEATED LOGIN FAILURES ON ttyp3 FROM
                intrepid.itstd.s, daemon
```

รูปที่ 11 ตัวอย่างของ /use/adm/messages

ใช้ su เป็น root ได้ รวมทั้งช่วยบอกด้วยว่ามีใครกำลังทาย password ของ account โดยการทดลองบ้าง เนื่องจากจะบอกว่ามีการ login ทั้งใส่ password ถูกและผิดกี่ครั้ง

SHOWMOUNT

จะช่วยบอกชื่อเครื่องที่ใส่ตัวเซอร์ฟเวอร์อยู่ ถ้าใส่ออปชัน -a จะบอกชื่อเครื่องและไดเรคทอรีทั้งหมด เช่นในรูปที่ 12

ถ้าใส่ออปชัน -d จะแสดงไดเรคทอรีทั้งหมดที่กำลังถูกเครื่องอื่นใส่ไปใช้อยู่ คำสั่ง showmount ช่วยให้เราดูได้ว่าเครื่องไหนกำลัง mount ไดเรคทอรีไหนของเราไป ถ้ามีเครื่องแปลกปลอมหรือใช้ไดเรคทอรีที่ไม่ควรให้ไป ก็จะได้ตรวจสอบกันได้บ้าง

```
bronto.itstd.sri.com:/usr/share
bronto.itstd.sri.com:/usr/local.new
bronto.itstd.sri.com:/usr/share/lib
bronto.itstd.sri.com:/var/spool/mail
cascades.itstd.sri.com:/sparky/a
clyde.itstd.sri.com:/laser_dumps
cm1.itstd.sri.com:/sparky/a
coco0.itstd.sri.com:/sparky/a
```

รูปที่ 12 ตัวอย่างผลลัพธ์ของคำสั่ง showmount

การตรวจสอบระบบไฟล์

การสอดส่องดูแลความปลอดภัยของระบบไฟล์นับว่าสำคัญที่เดียว ต้องดูว่ามีไฟล์ที่ PERMISSION ที่ไม่ควรเปิดให้กับผู้ที่ไม่ควรได้รับหรือไม่

FIND

คำสั่ง find เป็นคำสั่งมาตรฐานของยูนิกซ์คำสั่งหนึ่งที่ทรงพลังมาก สามารถใช้หาไฟล์ตามรูปแบบต่าง ๆ ของชื่อ ชนิด เจ้าของ วันเวลาแก้ไข ฯลฯ แล้วยังนำชื่อไฟล์ที่ได้ไปพิมพ์หรือไปเป็นอาร์กิวเมนต์ของคำสั่งอื่น ๆ ก็ได้ สำหรับประโยชน์ในการรักษาความปลอดภัย เช่น

- หาไฟล์ที่ SETUID หรือ SETGID โปรแกรมที่มีการ SETUID เป็น root จะต้องระมัดระวังตรวจสอบให้มาก เมื่อแฮ็กเกอร์สามารถเจาะระบบแล้วทำตัวเองเป็น root ได้แล้ว มักจะสร้างโปรแกรมที่ SETUID เป็น root ขึ้นมาทิ้งไว้ เพื่อเป็นช่องทางสำรองไว้ครา่วต่อไป คำสั่ง find ช่วยหาไฟล์เหล่านี้ได้ โดย

```
#find / -type f -a \( -perm -4000 -o -perm -2000 \) -print
```

โดย / หมายถึงให้เริ่มหาตั้งแต่ / หรือทุก ๆ ไฟล์นั่นเอง -type f หมายถึงให้ตรวจเฉพาะไฟล์ที่มีชนิดเป็น f หรือไฟล์ทั่วไป (regular file) ถ้าเป็น d จะหมายถึงไดเรคทอรี I หมายถึง symbolic link เป็นต้น

-a หมายถึง and เพื่อแสดงตรรกว่าต้องการเงื่อนไขทั้งสองแบบ

\(-perm -4000 -o -perm -2000 \) ทั้งหมดในวงเล็บนี้บอกความเป็นกลุ่มเดียว โดยที่ -perm -4000 หมายถึงไฟล์ที่บิต PERMISSION เซตบิต 4000 (ฐานแปด) ไว้ ซึ่งก็คือบิตของ SETUID ส่วน -o หมายถึง or เป็นการแสดงตรรกการเลือกอย่างใดอย่างหนึ่งหรือทั้งสองอย่าง และ -perm -2000 หมายถึงไฟล์ที่บิต PERMISSION เซตบิต 2000 (ฐานแปด) ไว้ ซึ่งก็คือบิตของ SETGID ดังนั้นนิพจน์นี้ทั้งหมดก็บอกว่าเซตบิต SETUID หรือ SETGID นั่นเอง

-print อันนี้บอกให้พิมพ์ชื่อของไฟล์ที่ตรงตามกฎเกณฑ์ที่กำหนดให้ออกมาทาง stdout หรือจอเทอร์มินัล

เมื่อสั่งรันคำสั่งนี้แล้ว (ด้วย root) สักประมาณ 10 นาทีหรืออาจจะถึงชั่วโมง (ขึ้นกับขนาดของดิสก์และจำนวนไฟล์ทั้งหมดที่มี) ก็จะแสดงผลลัพธ์ออกมาเป็นรายชื่อไฟล์ ซึ่งควรจะตรวจสอบว่าแต่ละไฟล์นั้นสมควรหรือไม่

- หาไฟล์ที่ทุกคนเขียนได้ ไฟล์ของระบบที่ทุกคนเขียนได้อาจจะเป็นช่องว่างให้แฮ็กเกอร์ก็ได้ อาจจะมาเพิ่มหรือลบไฟล์บางไฟล์ทิ้งได้ รูปแบบคำสั่ง find ที่ช่วยหาให้ได้ก็คือ

```
#find / -perm -2 -print
```

ในที่นี้เราไม่ใช้ -type เพราะเรากำลังจะสนใจทั้งไฟล์และไดเรคทอรี รวมทั้ง device ด้วย ส่วน -perm -2 นี้หมายถึง world-write bit นั่น

เอง พอคำสั่งนี้รันแล้วจะได้ผลลัพธ์ค่อนข้างมาก แต่บางไฟล์ก็ไม่ต้องสนใจ เช่น อุปกรณ์พวกเทอร์มินัลใน /dev หรือ error log ไฟล์หรือ symbolic link เนื่องจาก PERMISSION ใน symbolic link ไม่มีความหมายอะไร

- หาไฟล์ที่ไม่มีเจ้าของ ปกติไฟล์ประเภทนี้เกิดขึ้นตอนที่ลบ account ออกไป แล้วเหลือไฟล์ของคนนั้นอยู่ ซึ่งควรจะถูกลบออกไปด้วยรวมทั้งไฟล์ที่แฮ็กเกอร์อาจสร้างขึ้น ใช้คำสั่ง find หาโดย

```
# find / -nouser -o -nogroup -print
```

ออปชัน -nouser จะหาไฟล์ที่มี USER ID ที่ไม่ปรากฏในไฟล์ /etc/password ส่วนออปชัน -nogroup จะหาไฟล์ที่มี GROUP ID ซึ่งไม่ปรากฏในไฟล์ /etc/groups และใช้ -o เชื่อมเพื่อเลือกอย่างใดอย่างหนึ่งหรือทั้งสองอย่าง

- หาไฟล์ .rhosts อย่างที่กล่าวมาแล้วตั้งแต่ต้นว่าไม่ควรให้มีการใช้ไฟล์ .rhosts สำหรับการหาไฟล์ .rhosts ก็ทำได้โดย

```
# find /home -name .rhosts -print
```

เมื่อ /home เป็นไดเรคทอรีที่เก็บ home ไดเรคทอรีของผู้ใช้ทั้งหลายในระบบ

BACKUP

การทำสำรองข้อมูลนอกจากจะช่วยกู้ภัยเวลาเกิดความผิดปกติทางฮาร์ดแวร์ของดิสก์ หรือการเผลอลบไฟล์บางไฟล์แล้วยังช่วยป้องกันรักษาความปลอดภัยด้วย วิธีทำสำรองข้อมูลที่ดีควรจะ dump ทั้งระบบหรือ full dump อย่างน้อยเดือนละครั้ง และทำสำรองส่วนที่เพิ่มหรือ incremental dump สัปดาห์ละ 2 ครั้งหรือทุกวันได้ก็ดี คำสั่ง dump จะดีกว่า cpio หรือ tar เนื่องจาก dump จะสามารถคืนสภาวะของระบบไฟล์ได้เหมือนเดิมที่สุดตอน restore กลับ

CHECKLISTS

เป็นรายการทั้งหมดของไฟล์ในไดเรคทอรีกลุ่มหนึ่งซึ่งบอกขนาด เจ้าของ วันเวลาแก้ไขและอื่น ๆ ซึ่งใช้เปรียบเทียบกับที่เคยทำมาครั้งก่อน เพื่อตรวจสอบความเปลี่ยนแปลงได้ในระดับเบื้องต้น ปกติแล้วจะไม่ทำกับไดเรคทอรีที่เก็บบ้านของผู้ใช้ทั้งหลาย เนื่องจากภายใต้นั้นจะมีการเปลี่ยนแปลงเสมอ เมื่อตรวจสอบพบความเปลี่ยนแปลงในไฟล์ checklists ที่ทำครั้งนี้กับที่ทำครั้งที่แล้ว ก็เป็นหน้าที่ที่จะต้องตรวจสอบดูว่าสิ่งที่เปลี่ยนแปลงไปนั้นจะเป็นอันตรายต่อระบบหรือไม่

ปัจจุบันมีซอฟต์แวร์ที่ทำ checklists อย่างนี้ให้แล้ว แต่ถ้าจะทำเองด้วยคำสั่งมาตรฐานง่าย ๆ ของยูนิกซ์เองก็ได้คือคำสั่ง ls โดยควรทำต้นฉบับไว้ตั้งแต่เพิ่งลงระบบปฏิบัติการใหม่ ๆ หรือเมื่อใดก็ได้ที่มั่นใจว่าระบบยังปลอดภัยอยู่ ตัวอย่างการทำต้นฉบับหรือ master ก็คือ

```
# ls -aslgR /bin /etc /usr > MasterChecklist
```

แล้วก็อาจจะตัดหรือแก้ไขบางส่วนที่รู้ว่าต้องมีการเปลี่ยนแปลงแน่นอน เช่น /etc/utmp, /usr/adm/acct แล้วไฟล์ MasterChecklist นี้ต้องเก็บไว้ในที่ ๆ ปลอดภัยจริง ๆ ไม่ให้แฮ็กเกอร์มาแก้ไขได้เช่นเก็บขึ้นเทปไว้

พอใช้งานระบบไปสักพักหนึ่งก็มาตรวจสอบดูโดยคำสั่งเดียวกัน แต่ส่งผลลัพธ์ลงไฟล์ชื่ออื่น เช่น CurrentChecklist แล้วมาเปรียบเทียบ 2 ไฟล์นี้ เช่นด้วยการใช้คำสั่ง diff

```
# diff MasterChecklist CurrentChecklist
```

ก็จะรู้ได้ว่ามีอะไรหายไป เกินมา หรือเปลี่ยนแปลงไป ถ้าแน่ใจว่าสิ่งที่เปลี่ยนแปลงนั้นไม่มีอันตรายอะไรก็อาจปรับปรุงไฟล์ MasterChecklist โดยการก็อบปี้จาก CurrentChecklist ก็ได้

ตารางสรุป

ท้ายนี้จะเป็นตารางในรูปที่ 13 ที่สรุปข้อมูลเกี่ยวกับความปลอดภัยในระบบปฏิบัติการยูนิกซ์ซึ่งสามารถใช้ตรวจสอบได้

หนังสืออ้างอิง

1. David A. Curry. "Improving the Security of Your UNIX System." SRI International. April 1990.

2. Eichin, Mark W., and Jon A. Rochlis. With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988. Massachusetts Institute of Technology. February 1989.

3. F. T. Grammp and R. H. Morris. "UNIX Operating System Security." AT&T Bell Laboratories Technical Journal. October 1984.

4. H.M. Deitel. "Operating System." Second edition. Addison Wesley. February 1990.

5. Morris, Robert, and Ken Thompson. "Password Security: A Case History." Communications of the ACM, 22 (11): 594-597, November 1979. Reprinted in UNIX System Manager's Manual, 4.3 Berkeley Software Distribution. UC. Berkeley. April 1986.

6. Seeley, Donn. A Tour of the Worm. Department of Computer Science, University of Utah. December 1988.

7. Stoll, Clifford. "Stalking the Wily Hacker." Communications of the ACM, 31 (5): 484-497, May 1988.

```
Account Security
     []  Password policy developed and distributed to all users
     []  All passwords checked against obvious choices
     []  Expiration dates on all accounts
     []  No ``idle'' guest accounts
     []  All accounts have passwords or ``*'' in the password field
     []  No group accounts
     []  ``+'' lines in passwd and group checked if running Yellow Pages

Network Security
     []  hosts.equiv contains only local hosts, and no ``+''
     []  No .rhosts files in users' home directories
     []  Only local hosts in ``root'' .rhosts file, if any
     []  Only ``console'' labeled as ``secure'' in ttytab (servers only)
     []  No terminals labeled as ``secure'' in ttytab (clients only)
     []  No NFS file systems exported to the world
     []  ftpd version later than December, 1988
     []  No ``decode'' alias in the aliases file
     []  No ``wizard'' password in sendmail.cf
     []  No ``debug'' command in sendmail
     []  fingerd version later than November 5, 1988
     []  Modems and terminal servers handle hangups correctly

File System Security
     []  No setuid or setgid shell scripts
     []  Check all ``nonstandard'' setuid and setgid programs for security
     []  Setuid bit removed from /usr/etc/restore
     []  Sticky bits set on world-writable directories
     []  Proper umask value on ``root'' account
     []  Proper modes on devices in /dev

Backups
     []  Level 0 dumps at least monthly
     []  Incremental dumps at least bi-weekly
```

รูปที่ 13 ตารางสรุปสิ่งที่ควรตรวจสอบเพื่อความปลอดภัยของระบบปฏิบัติการยูนิกซ์